TEXAS A&M UNIVERSITY
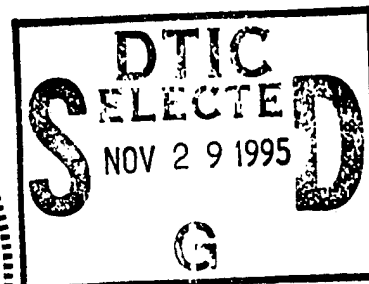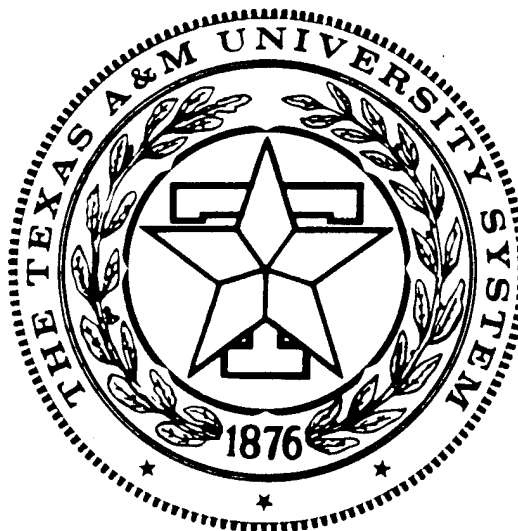
Hardware Implementation of a Desktop Supercomputer for High Performance Image Processing

ONR Grant Number N00014-94-1-0516

Technical Report
Aug/01/95 – Nov/01/95

S

DTIC
SELECTED
NOV 2 9 1995
G

DEPARTMENT OF ELECTRICAL ENGINEERING
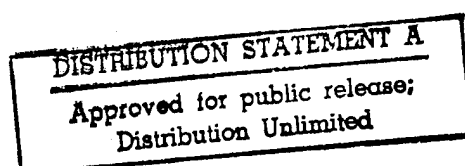### College Station, Texas

Hardware Implementation of a Desktop Supercomputer for High Performance Image Processing

ONR Grant Number N00014–94–1–0516

Technical Report
Aug/01/95 – Nov/01/95

DTIC SELECTED NOV 2 9 1995 G

19951122 079

Time Multiplexed Color Image Processing based on a VLSI Cellular Neural Network with Cell–State Outputs
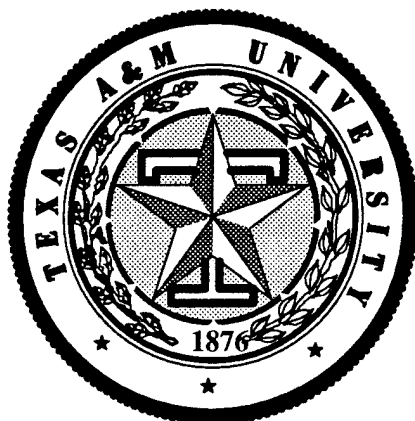
Dr. Jose Pineda de Gyvez

Texas A&M University
Microelectronics Group
Department of Electrical Engineering
College Station, TX, 77843

Phone: (409) 8457477
FAX: (409) 8457161
Email:gyvez@pineda.tamu.edu

1. Introduction

Since their introduction in 1988, Cellular Neural Networks have shown a vast computing power, especially for image processing [1,2,3,4]. Many VLSI implementations of CNN analog neural networks have been proposed in recent years [15,18,23]. These implementations include OTA based processing elements [8], discrete time implementations [5,10], switch–current signal processing elements [14], and the current–mode [13] implementation. Each kind of CNN realization has its own advantages and disadvantages. For example, the discrete–time CNN can yield "exact" template weights and RC–constant, but it takes more area and consumes more power [11,13].

Early CNN implementations were designed to perform one function in image processing or classification, such as edge detection [2,3], connected component detection [11], or hole filling. More recently programmability, i.e. the ability to change template values electronically, has been studied in detail [6,7,10,23,24]. Furthermore, in some implementations the activation function of some CNN chips is also tunable e.g. [16,17], the slope and the threshold of the activation function.

One common feature of currently available CNN circuits is that the output signals are the feedback outputs of the cells, and those output values are confined as binary values [1]. Hence, the output image is a black and white image even when the CNN, in its nature, is an analog and continuous signal processing system. The binary output values of the CNN are the positive or negative threshold of the activation function. Due to this non–linear sigmoid function, the feedback output of a cell can converge to either a positive or negative value under some well studied "stability conditions," such as $A_{ii} > 1$. This characteristic makes the CNN very attractive for some pattern extracting applications, such as edge detection and connected element detection, where a binary valued output image is acceptable. Moreover, the circuit design is relatively easy if the output is just binary rather than continuous, since the linearity, precision, and offsets of the output values are not relevant [9,13,15,23] because the steady state is not of importance. Worth mentioning is that the silicon area and power dissipation are greatly reduced because of

the tradeoffs between precision and area, or power dissipation.

However, in some cases, the binary output CNN is not enough. For example, in order to solve a group of differential equations, or to build a real time control system, or to obtain an output image with multiple gray levels (color levels), a CNN with linear continuous outputs is required. Since the feedback outputs are limited by thresholds, e.g. -1 and 1, state variables have relatively wider dynamic ranges than the feedback outputs and therefore can be used as continuous outputs. In some CNN theoretical research papers [25], the state variable (or state output) has already been mentioned as a useful continuous information of the CNN. Some authors also define state variables as the roots of the differential equations and hence solve differential equations with the CNN.

So far, except for the above fundamental work, there are no circuits that have been fabricated or designed for the purpose of obtaining continuous state outputs. Although in some CNN chips the state variables could be detected [13,18], they were not used as outputs. Some major electrical problems raised by the design of such a CNN chip are summarized below:

1. The linearity of each circuit block, including the multipliers (associated with template values), linear resistor, and the activation function (piecewise linear sigmoid function) must be seriously considered. Any distortion in those blocks will contribute to the non–linear error of the state variable output.

2. The dynamic range of the state variable is bounded by the linear range of circuit blocks or the power supply voltage. In order to get a high precision and resolution output signal, a wide swing range is needed, but the area and power dissipation of the chip have to be increased in turn.

This paper addresses two key aspects in the field of Cellular Neural Networks. One is the development of a monolithic prototype (a 3 × 3 CNN array) that uses the state node as an external output for gray level processing. The second aspect is the integration of this IC in a complex system. It is necessary to stress out that the state of the art

2

work in Cellular Neural Networks has concentrated on VLSI implementations without really addressing the "systems level". While efficient implementations have been reported, no reports have been presented on the use of these implementations for processing large complex images. The work hereby presented introduces a strategy to process large images using small CNN arrays. The approach, time–multiplexing, is prompted by the need to simulate hardware models and test hardware implementations of CNN. For practical size applications, due to hardware limitations, it is impossible to have a one–on–one mapping between the CNN hardware processors and all the pixels in the image involved. This paper presents a practical solution by processing the input image block by block, with the number of pixels in a block being the same as the number of CNN processors in the hardware.

2.1 System Structure

The CNN IC consists of a 3 × 3 array with shared input/output pins. Salient features of this implementation are full template programmability, a programmable RC integration constant and an external output at the state node.

Figure 1 presents a modular view of the CNN IC along with I/O signals.

- $b_{11}, b_{12}, \cdots, b_{33}$ are the pins to set the analog template values of B_{ij}, where $i, j = 1, 2, 3$.

- $a_{11}, a_{12}, \cdots, a_{33}$ are the pins to set the analog template values of A_{ij}, where $i, j = 1, 2, 3$.

- IO1, IO2, $\cdots$, IO9 are the input–output pins of all nine cells. The pin of each cell is used to do the functions of setting the boundary conditions, initializing the state, and of providing external input values to the cell, as well as obtaining its state output.

- $d1$ and $d2$ are control signals to multiplex each input–output pin for different functions at different time periods.

- V_{bias} is the offset bias voltage for the templates, and V_c is a tuning voltage of the active resistor.

- 5V, -5V, 1V, -1V are power supplies for the circuit and for the function activation, respectively.

Two control signals are used as switching signals to multiplex inputs, outputs, and cell initial conditions in order to let them use the same pins. Data lines are shared by analog inputs, boundary values, initial conditions of cell state variables, and outputs of state variables. The logic codes and sequences of pin multiplexing is shown in Table 1.

Table 1: The codes and sequence of pin multiplexing operations.

sequence	code	operation
1	11	set boundary values (in)
2	10	set initial conditions (in)
3	01	set input voltages (in)
4	00	extract the steady-state outputs

The pin multiplexing scheme is shown in Figure 2. Capacitors (0.6pF) are added to hold the input information when the circuit is switched to the output mode. This capacitance value is designed to eliminate the feed-through effects of CMOS switches, and for the same purpose, all analog switches are transistors with minimal size. As a result, the output is kept unchanged when the pin is switched from the input to the output voltage. The terminal to set the initial condition is connected to the state variable node of the cell.

2.2 Cell Core

Figure 3 presents the hardware realization of a CNN cell. Here the RC integrator is composed of an opamp and an OTA which is in the feedback path. The OTA is used to act as an active resistor , with a value $g_x = 1/R_x$. The purpose of adding the opamp is to isolate the RC integrator from the 19 multipliers used to implement the weights of both A and B templates.

Observe that when the 19 multipliers are connected in parallel a much smaller net output resistance than that of just one multiplier (divided by 19), and also a much larger

net parasitic capacitance than that of one multiplier (times 19) appear. These two non-ideal elements could reduce the value of R_x and increase the value of C_x in the structure of Figure 3 because of their parallel connections with each other. However, the *virtual ground point* (non-inverting input) of the opamp can isolate the output impedances of the multipliers from R_x and C_x. On the other hand, the virtual ground makes each multiplier have a "virtual" zero load, and thus eliminates the load effect on the multiplier which comes from the finite output impedance of the transconductance multiplier.

Another advantage of isolating the large aggregated parasitic capacitance of 19 multipliers is that the value of C_x can be controlled by a single capacitor, rather than by many parasitic values. In this way, it is possible to control the time constant of the RC integrator. More importantly, the mismatches of the RC time constants of the cells are much smaller if the value of C_x is decided by one monolithic capacitor rather than by many distributed parasitic values.

To prevent the RC integrator from entering into oscillation, the value of C_x has to be large enough to form a dominant pole, and thus compensate for the phase shift of the integrator. Assuming the opamp is an ideal opamp and without C_x, the transfer function of the integrator can be expressed as:

$$\frac{v_x(s)}{v_{in}(s)} = \frac{v_{in}(s)G(s)}{g_x(1 - p_1 s)(1 - p_2 s)}, \tag{1}$$

where $G(s)$ indicates the transconductance of a multiplier (to simplify the presentation, assume just one multiplier); g_x is a constant; p_1 and p_2 are two zeros of the OTA, but here they become poles. In Equation 1, the OTA is simplified as a 2nd order system. The two poles, p_1 and p_2, are possible factors to cause oscillations depending on the total phase shift. The most convenient way to stabilize this transfer function is to choose an adequate value of C_x, and thus compensate the phase shift. After adding C_x, Equation (1) is written as:

$$\frac{v_x(s)}{v_{in}(s)} = \frac{G(s)}{g_x\left[(1 - p_1 s)(1 - p_2 s) + C_x s/g_x\right]} \tag{2}$$

In Equation 2, if $C_x/g_x >> p_1, p_2$, there will be a dominant pole based upon the ratio C_x/g_x that makes the integrator be stable.

Finally, the buffer in Figure 3 is basically an opamp with unity-gain feedback connection whose function is to isolate the state variable node of the cell from the outside environment.

2.3 Multiplier Circuit

For the design of a continuous output CNN, the general requirements for the transconductance multiplier are the linearity and its tolerance to process mismatches. After evaluating several analog multipliers, the best choice was found to be the Gilbert transconductance structure, which has been extensively used in the design of analog neural networks. The Gilbert multiplier has the advantage that its linearity is relatively insensitive to process mismatches because of its symmetric structure and differential input and differential output. The tradeoffs of the above good news is having to enlarge the layout area and increase its power dissipation to get satisfactory results.

The multiplier used here is a folded Gilbert multiplier with linearity enhancement, which is illustrated in Figure 4. PMOS transistors M1 and M2 form a current mirror pair to supply a biasing current to the input pair M3 and M4, thus $I_5 = I_6 = I_b/2$. M3 and M4 are biased in their linear regions as source degenerated resistors, whose functions are to expand the linearity of the input pair M5 and M6. V_c is a control voltage that represents the template value. In the following equations, M3 and M4 are ignored for convenience of the mathematical analysis of the multiplier principle.

It can be proven that the output differential current of this multiplier is

$$
\begin{aligned}
I_{o1} - I_{o2} &\approx k'V_{in}(\sqrt{2I_9/k'} - \sqrt{2I_{10}/k'}) \\
&= k'V_{in}\sqrt{2k/k'}(V_{gs5} - V_{gs6} + V_{thP} - V_{thP}) \\
&= \sqrt{2kk'}V_{in}V_c.
\end{aligned}
\tag{3}
$$

where $k = k_5 = k_6 = \frac{1}{2}\frac{W}{L}K_p$ and $k' = k_{11} = k_{12} = k_{13} = k_{14}$

Equation 3 is the fundamental equation to perform four-quadrant multiplications with the input voltage V_{in} and the control voltage V_c. The linearity of the circuit can be improved by either using long channel transistors or large biasing currents, since the value of $\sqrt{2I_{10}/k'}$ can be much larger than V_{in}.

In order to save area and power dissipation, a summing current mirror (shown in

7

Figure 5) is utilized to collect the output currents of all 19 multipliers instead of adding individual current mirrors for every multiplier at their output stage. This cascode summing current mirror is needed to minimize output offsets and make the circuit properties insensitive to process variations. Table 2 shows the designed circuit parameters. HSPICE simulations showed a total harmonic distortion (THD) of 0.5%, at $V_{in} = 1.0V$ and $V_c = 1.0V$, and a power dissipation of about 0.75mW using the process parameters of $2\mu m$ n–well CMOS technology from MOSIS.

Table 2: Design parameters of the Gilbert multiplier in Figure 4 ($I_b = 10\mu A$).

transistor	$W/L(\mu m/\mu m)$
M1 & M2	4/5
M9 & M10	9/4
M3 & M4	3/7
M11, M12, M13, & M14	3/26
M5 & M6	3/22
M7 & M8	8/4

2.4 The Linear Tunable OTA

The linear resistor of the cell core was implemented with a tunable OTA. The tunability of the OTA is also indispensable because 1) on–chip tuning is required to compensate systematic errors raised by parameter mismatches, and 2) CNN *programmability* involves that the value of R_x be also adjustable. The traditional method of tuning the transconductance of the OTA is by adjusting the biasing current of the input differential pair. The linearity is significantly reduced by decreasing its biasing current because of the quadratic relationship between the gate–source voltage and the drain current. Hence, this method is not suitable for our purposes. One more adequate way to simultaneously improve both the linearity and tunability of the OTA is to utilize a *programmable current mirror* [20,21]. This approach has been found to be very good for both linearity and tunability. It consists of adjusting the gain of the current mirrors rather than the biasing current of the input pair. In our case, a modified programmable current mirror (programmable Widlar current

mirror) is presented. It has a simple structure and its performance is also good.

2.4.1 Circuit Analysis

The circuit structure of the OTA is shown in Figure 6. The only difference of this circuit from a *basic* CMOS OTA is that there are two transistors MR1 and MR2, which are biased in their linear regions. The functions of both active resistors are to tune the current gains of two current mirrors: M3–M5 and M4–M6, as well as to increase the linear range.

Let the linear resistance of MR1 and MR2 be denoted as R, $k \triangleq k_1 = k_2$, $k \triangleq k_3 = k_5$ and let $V_d \triangleq V_{in+} - V_{in-}$.

It is reasonable to assume the values of k and k' to be larger than $10^{-5} V^2/A$, R larger than $100k\Omega$, and the biasing current I_b large enough (e.g., 35uA). Then within a limited input range such that I_3 and I_4 are not far from $I_b/2$, we have

$$4R\sqrt{k'(I_3)} >> 1 \tag{4}$$

$$4R\sqrt{k'(I_4)} >> 1 \tag{5}$$

Therefore, by obtaining I_5 and I_6 in terms of I_3 and I_4, respectively around MR1 and MR2 the output current can be approximated as

$$I_{out} = I_5 - I_6 = \frac{1}{2k'R^2}[2R\sqrt{kk'}V_d - [2R^{1/2}(k'I_3)^{1/4} - 2R^{1/2}(k'I_4)^{1/4}] \tag{6}$$

In order to separate the linear and non-linear terms of I_{out}, it is better to expand it into polynomial expressions in terms of the differential input voltage V_d. All even terms vanish if it is assumed that the input voltage is differential. Disregarding high order terms we have

$$I_{out} = \alpha_1 V_d + \alpha_3 V_d^3 + \cdots,$$

where

9

$$\alpha_1 = \frac{dI_{out}}{dV_d}\Big|_{V_d=0} \tag{7}$$

$$\alpha_3 = \frac{1}{3!}\frac{d^3 I_{out}}{dV_d^3}\Big|_{V_d=0} \tag{8}$$

$$\vdots$$

Evaluating the corresponding derivatives, equations 7 and 8 yield respectively

$$\alpha_1 = \frac{1}{R}\sqrt{\frac{k}{k'}}[1 - \frac{1}{2}k'^{-5/4}R^{-1/2}(\frac{I_b}{2})^{-1/4}], \tag{9}$$

$$\alpha_3 \approx \frac{1}{12}R^{-3/2}k'^{-3/4}k^{3/2}(\frac{I_b}{2})^{-13/4}. \tag{10}$$

$$\vdots$$

The linear term in Equation 9 is the conductance of the OTA. In certain cases, such as when R is large enough, α_1 can be further simplified as

$$\alpha_1 \approx \frac{1}{R}\sqrt{\frac{k}{k'}}. \tag{11}$$

Simultaneously notice from Equation 10, that the non–linearity (3rd order distortion) of I_{out} can be greatly reduced by increasing the values of R and I_b. Under the assumptions made in Equations 4 and 5, the 3rd order term is much smaller than the linear term.

Now recall that since MR1 and MR2 are working in the linear range, their equivalent resistance is

$$R = \frac{1}{k_R(V_{gs} - V_{thP})} \tag{12}$$

Then, it can be concluded that

$$I_{out} \approx k_R(\mid V_{dd} - V_t \mid - \mid V_{thP} \mid)V_d\sqrt{\frac{k}{k'}} \tag{13}$$

Another advantage of this programmable current mirror is that its gain bandwidth product will not change with the adjustment of its conductance. This point is easy to

understand since the biasing current of the input pair I_b is not changed. The design parameters of the OTA of Figure 6 are listed in Table 3.

Table 3: The design parameters of the OTA with programmable Widlar current mirror in Figure 6 ($I_b = 25\mu A$).

transistor	W/L ($\mu m/\mu m$)
M1 & M2	3/35
M3 & M4	4/4
M5 & M6	4/5
MR1 & MR2	3/19
M7, M8, M9 & M10	4/16

The OTA's input range is from $-3V$ to $3V$. The tuning range of the conductance is from $2 \times 10^{-6}\Omega^{-1}$ to $5 \times 10^{-6}\Omega^{-1}$. The simulated THD vs. input voltages is plotted in Figure 7. Although there are differences between different input voltages, these variations are within the tolerable range of linearity. The estimated total power dissipation of the OTA is 0.45 mW.

2.5 The Activation Function

The sigmoid activation function plays a very important role to control the errors and the stability of the CNN [19]. The most important aspects to consider are

1. The threshold voltages of the sigmoid function must be accurate to avoid measurement errors applied to the output voltage.

2. The slope of the linear segment should be the same among all cells in the CNN. Any mismatch may introduce stability problems.

3. The slope of the sigmoid function of each cell should be about 1.0 to make the cell more stable than other values.

4. The slew rate of the output voltage should be high to have a short settling time.

11

The circuit structure of the sigmoid activation function is shown in Figure 8. It is basically an opamp in unity gain–feedback connection, but the power supplies of the first and second stages of the opamp are different. The supply voltages of the first stage are −5 and +5 volts, while the voltages of the second stage are only −1 and +1 volts. All power voltages are supplied external to the IC. The advantage of using independent power supplies in one opamp is that the threshold voltages of the sigmoid function are well defined and programmable. Therefore we do not have to use hard limiter circuits, whose threshold voltages are always significantly variable with process variations.

The structure of the feedback connection of a high–gain opamp guarantees that the slope of the input–output characteristic curve is almost ideal 1.0 and that there are sharp turning corners at the points of $V_{in} = -1.0V$ and $V_{in} = 1.0V$.

These conditions make the active function be a perfect piecewise linear sigmoid function. However, the deep feedback connection of the opamp may introduce stability problems. A compensation capacitor C_c has to be added to compensate for the phase shift, but C_c will contribute to the time delay of the activation function. Within one chip, it is acceptable to assume that the relative mismatch (or the variation of the ratio) of the C_cs between two cells is very small, so the time delay mismatch caused by C_c is not critical. The simulated slew rate of the activation function is about $10V/\mu s$, at the load capacitance of $1pF$. The total power dissipation of the activation function is 0.38 mW. Table 4 lists the design parameters of the activation function.

Table 4: The design parameter of the activation function circuit ($I_b = 15\mu A$) .

transistor	W/L ($\mu m/\mu m$)
M1 & M2	4/4
M3, M4, M5 & M6	5/4
M7 & M8	7/4

2.6 Opamp

There are two opamps in each cell; one is in the cell core; another one is used as a buffer to isolate the parasitic capacitance of the outside world from the state variable node. Both opamps have a one–stage opamp structure as shown in Figure 9. The advantage of using a one–stage opamp is that it is more stable than a two–stage opamp, and the load capacitance does not affect the stability of the opamp. The gain of the one–stage opamp is not very high, but is enough for our applications. Its design parameters and some important HSPICE simulation results are listed in Table 5 and Table 6, respectively.

Table 5: The design parameter the opamp shown in Figure 9, $I_b = 20\mu A$.

transistor	W/L $(\mu m/\mu m)$
M1 & M2	4/4
M3, M4, M5 & M6	5/4
M7, M8, M9 & M10	5/4
M11, M12, M13 & M14	4/7

Table 6: HSPICE simulated results of the one-stage opamp in Figure 9.

parameter	result
DC gain	65 dB
Dynamic swing range	−3 to +3V
phase margin	50°
gain bandwidth product	4.0 MHz $(C_L = 1pf)$
slew rate	$10V/\mu s$
power dissipation	0.42mW

3. TIME–MULTIPLEXING HARDWARE SIMULATION

In time–multiplexing hardware simulations one can define a block of pixels (subimage) which will be processed by an equal number of CNN cells [22]. Once convergence is achieved, a new subimage adjacent to the one just processed, is scheduled for further processing. This procedure is repeated until the whole image has been scanned using a

lexicographical order, say, from let to right and from top to bottom. It is obvious that with this approach the processing of large images becomes feasible in spite of the finite number of CNN cells.

Even though the approach seems simple and appealing, an important observation is necessary: the processed border pixels in each subimage may have incorrect values since they are processed without neighboring information. Hence, to cope with the previous problem, two sufficient conditions must be considered to ensure that each border cell properly interacts with its neighbors. These conditions are: 1) to have a belt of pixels from the original image around the subimage being processed, and 2) to have pixel overlaps between adjacent subimages. We will go into the details of these two constraints in the next subsection.

3.1 Sufficient and Necessary Conditions for Time Multiplexing

Notice first that in the absence of template -A values the error is both image and template -B dependent. In other words, the steady state of a border cell may converge to an incorrect value due to the absence of it's neighbors weighted input. One can easily conclude that the error is canceled if the missing external inputs are provided to the border cells as depicted in Fig. 10a. Since typically, the array is "embedded" in the image during operation, this condition can easily be satisfied.

Let us address now the interactions among cells. The problem in this situation is more involved because the output signals depend on the state of their corresponding cells. To minimize the error an overlap of pixels between two adjacent blocks is proposed, see Fig. 10b. In this form, the inner cells of the CNN array will always receive weighted processing information from the border cells.

The general time–multiplexing procedure consists in processing each image block until all CNN cells within the block converge. The block with converged cells will have state output variables which are the values used for the final output image. Every time that a new subimage is processed, the physical CNN array is initialized to the initial conditions

of the original image, or to black or to white as required by the template in use. In the overlapping procedure the outer overlapped cell's converged values are discarded since they were computed with incomplete neighboring information. Only the inner cell's converged values are kept as valid values. This implies that for a neighborhood radius of 1, an overlap of two pixel column/rows is needed to be able to ensure correct values for pixels assigned to the border cells. With the added overlapping feature, better neighboring interactions are achieved, but at the same time, an increase in computation time is inevitable.

With the previous multiplexing scheme the image needs to be iterated several times over newly obtained states to allow the proper propagation of global effects. Multiple iterations are necessary to guarantee that all cells have converged to correct values taking into account all global effects. This can be inferred by considering a diagonal propagation of, say, a black pixel in a fully white image. Notice that without overlaps it is impossible to propagate global effects, and that the propagation is achieved with at least one over lapped pixel.

For the purpose of better understanding the overall idea of this time–multiplexing approach, a simplified algorithm is presented below. Assume and M by N image, an m y n CNN array, pixel values E_{ij} and o overlaps.

$$\text{for} \quad (i = 1;\ i \leq M;\ i = i + m - o) \ \{$$
$$\quad \text{for} \quad (j = 1;\ j \leq N;\ j = j + n - o) \ \{$$
$$\quad u(i,j) \ = \ E_{i,\, j}$$
$$\quad x_{i+m,j+n}(t_o) = \begin{cases} u_{ij} \\ \text{black} \\ \text{white} \end{cases}$$
$$\quad x_i + m,\ j + n(T_{n+1} = x_{i+m,\ j+n}(t_n) \ + \ \int_{t_n}^{t_n+1} f'(x_i + m,\ j + n(t_n))\ dt$$
$$\quad \{$$
$$\{$$

4.1 Testing of Electrical Parameters

The 3×3 CNN chip was fabricated with MOSIS n–well $2.0\mu m$ process. The photograph of the die is shown in Figure 11a, where all cells are arranged as a 3 by 3 array. The die area of the circuit is approximately $3.2mm^2$, and the power dissipation is less than 250 mW. The photograph of one cell is in Figure 11b, whose layout area is $0.19mm^2$.

Two important circuit building blocks; the analog multiplier and the tunable linear OTA, were also fabricated in separate chips for testing purposes. The DC sweep curves of the OTA at different values of V_c are shown in Figure 12. The DC sweep characteristics of the multiplier are shown in Figure 13.

The linearity of the central cell $C(2,2)$ can be calibrated by adjusting the g_m of the tunable OTA using the procedure previously described. It is necessary here to deduct an amount of $-0.51V$ from V_{bias}, since this value is used to cancel the output offset of $C(2,2)$ and cannot be counted in the calculation of linearity.

The CNN chip is connected to a personal computer (PC) through an A/D and a D/A interfacing board. The system connections are described in Figure 14. The operations of setting inputs and getting outputs from the CNN chip are multiplexed externally by 4–1 analog multiplexer chips (ADG509A). External operations are synchronized with the multiplexing operations inside the chip. The type of A/D board was AT–MIO–16F, which has 12 A/D channels; the type of D/A board was AT–AO–6/10, which has 10 D/A channels. Both are products of National Instruments. The pin multiplexing control code, is generated by a computer program and interfaced through the digital I/O port in the AT–MIO–16F board.

The detailed connections of the analog multiplexers are shown in Figure 15, where the opamps are added as A/D output buffers to isolate the output node from the parasitic capacitance of the wires and the A/D board.

The operating sequence is listed as below:

16

1. Initialize the A/D and D/A boards. Set the required template values by providing the corresponding analog voltages for the template values.

2. Set the boundary values of the 3 × 3 CNN, and the initial values of all the CNN cells.

3. Map the pixel values (0 to 255) of the input image into CNN input voltage values ($1.0V$ to $-1.0V$), and send them to the chip.

4. Extract the steady–state values ($3.0V$ to $-3.0V$) of the state variables of all cells and map them to pixel values (0-255) of the output image.

5. For time multiplexing applications [22], move to another position in the input image and repeat at step 2.

Another import function of tuning g_m is to expand the adjustable range of template values. For example, reducing g_m can increase the value of $B(i,j)$ ($= G/g_m \times b_{ij}$).

4.2 Image Processing Applications

The following comprises several examples of image processing applications using this 3 × 3 CNN chip, with output pixels at the state outputs. The size (number of pixels) of the input and output image are 256 × 256. The number of gray levels is 255. We chose a gray level image and a color image as demonstration vehicles of the potential of CNN state output nodes, see Figs. 16a and 16b.

The image processing is realized by using the *time multiplexing* method. Each time the CNN chip only processes a 3 × 3 pixel array of the image, but the border cells of the CNN are overlapped between the two neighbor arrays to have correct boundary dynamics. Therefore, only cell, $C(2,2)$ gives the output pixel value.

4.2.1 Edge Detection

The edge detection templates are the following [2]:

	B	
0	–.48	0
–.48	2.0	–.48
0	–.48	0

	A	
0	0	0
0	2.0	0
0	0	0

$$V_{bias} = -0.15V$$

The processed black and white output image for the template at $A(i,j;i,j) > 1/R_x$ is displayed in Figure 17a. The color results are shown in Fig. 17b. In our experiments, the value of V_{bias} affects the results very much. In order to obtain a good edge, several adjustments of V_{bias} may be needed.

4.2.2 Hole Filling

The hole filling function can be used in contrasting operatons and noise removal. This function is realized by the mutual feedback of output pixel values.

	B	
0	0	0
0	2	0
0	0	0

	A	
0	1.0	0
1	2	1
0	1	0

$$V_{bias} = -0.85V$$

The corresponding output images are in Figures 18a and 18b.

5. Conclusions

This paper demonstrated the feasibility of processing large images using small CNN arrays. For practical image size applications, due to current state of the art technological limitations, it is impossible to have a one–on–one mapping between the CNN hardware cells and all the pixels in the image involved. It is thus a key issue the proper use of time–multiplexing implementations in common–day situations. Additionally, it was shown

18

that a state–node output approach is especially suitable for color image processing. This approach is not limited by the constraint $A_{ii} < 1$ in order to obtain analog values.

REFERENCES

[1] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits & Systems*, vol. CAS–35, pp. 1257–1272, 1988.

[2] L. O. Chua and L. Yang, "Cellular neural networks: Applications," *IEEE Trans. Circuits & Systems*, vol. CAS–35, pp. 1273–1289, 1988.

[3] L. O. Chua and T. Roska, "The CNN paradigm," *IEEE Trans. Circuits & Systems*, vol. CAS–40, pp. 147–155, 1993.

[4] T. Roska and L. O. Chua, "The CNN universal machine: An analogic array computer," *IEEE Trans. Circuits & Systems*, vol. CAS–40, pp. 163–172, 1993.

[5] J. A Nossek and G. Seiler, "Cellular neural networks: Theory and circuit design," *Int. J. Circ. Theory Appl.*, vol. 20, pp. 533–553, 1992.

[6] G. F. Betta, S. Graffi, Z. M. Kovács, and G. Masetti, "CMOS implementation of an analogically programmable cellular neural network," *IEEE Trans. Circuits & Systems, Pt. II*, vol. CAS–40, pp. 206–215, 1993.

[7] P. Kinget and M. Steyaert, "A Programmable Analog Cellular Neural Network CMOS Chip for High Speed Image Processing," *IEEE Journal of Solid–State Circuits*, vol. 30, No. 3, pp. 235–243, March 1995.

[8] T. Kacprzak and K. Slot, "Multiple–input OTA based circuit for cellular neural network implementation in VLSI CMOS technology," *Proc. IEEE CNNA–92*, pp. 157–162, 1992.

[9] A. Rodríguez–Vázquez, R. Domínguez–Castro, and J. L. Huertas, "Accurate design of analog CNN in CMOS digital technologies," *Proc. IEEE CNNA–90*, pp. 273–280,

1990.

[10] M. Anguita, F J. Pelayo, A. Prieto, and J. Ortega, "Analog CMOS implementation of a discrete time CNN with programmable cloning templates," *IEEE Trans. Circuits & Systems, Pt. II*, vol. CAS–40, pp. 215–218, 1993.

[11] T. Matsumoto, L. O. Chua, and H. Suzuki, "CNN cloning template connected component detector," *IEEE Trans. Circuits & Systems*, vol. CAS–37, pp. 633–638, 1990.

[12] F. Sargeni, V. Bonaiuto, "High performance digitally programmable CNN chip with discrete templates," *Proc. IEEE CNNA–94*, pp. 67–72, 1994.

[13] J. E. Varrientos, E. Sánchez–Sinencio, and J. Ramírez–Angulo, "A current–mode cellular neural network implementation," *IEEE Trans. Circuits & Systems, Pt. II*, vol. CAS–40, pp. 147–155, 1993.

[14] S. Espejo, A. Rodríguez–Vázquez, R. Dom'ınguez–Castro, and J. L. Huertas, "Switched–current techniques for image processing cellular neural networks in MOS VLSI," *Proc. IEEE Int. Symp. Circuits & Systems*, pp. 1537–1540, 1992.

[15] J. L. Huertas, A. Rodríguez–Vázquez, and S. Espejo, "Analog VLSI implementation of cellular neural networks," *Proc. CNNA–92*, pp. 141–150, 1992.

[16] B. J. Sheu, S. H. Bang, and W.–C. Fang, "Optimal solutions of selected cellular neural network applications by the hardware annealing method," *Proc IEEE CNNA–94*, pp. 279–284, 1994.

[17] B. J. Sheu, s. H. Bang, and W.–C. Fang, Analog VLSI design of cellular neural networks with annealing ability," *Proc. IEEE CNNA–94*, pp. 387–391, 1994.

[18] J. M. Cruz and L. O. Chua, "Design of high–speed, high–density CNNS in CMOS technology," *Int. J. Circ. Theory Appl.*, vol. 20, pp. 555–572, 1992.

[19] K. Halonen, J. Väänänen, "The non–idealities of the IC–realization and the stability

of CNN–networks," *Proc. IEEE CNNA–90*, pp. 226–234, 1990.

[20] W. J. Adams and J. Ramírez–Angulo, "Extended transconductance adjustment/linearisation technique," *Electronics Letters*, vol. 27, pp. 842–844, 1991.

[21] J. Ramíez–Angulo and I. Grau, "Wide g_m adjustment range, highly linear OTA with linear programmable current mirrors," *Proc. IEEE Int. Symp. Circuits & Systems*, vol. 3, pp. 1372–1375, 1992.

[22] C.–C. Lee and J. Pineda de Gyvez, "Time–multiplexing CNN simulator," *Intl. Symp. on Circuits & Systems*, pp. 407–410, London, May 1994.

[23] A. Rodíguez–Vázquez, R. Domínguez–Castro, and S. Espejo, "Design of CNN universal chips: trends and obstacles," *Proc. CNNA–94*, pp. 59–60, 1994.

[24] J. M. Cruz, L. O. Chua, "A fast, complex and efficient test implementation of the CNN universal machine," *Proc. IEEE CNNA–94*, pp. 61–66, 1994.

[25] C.–C. Lee and J. Pineda de Gyvez, "Color Image Processing in a Cellular Neural Network Environment," *IEEE Trans. on Neural Networks*, to be published.
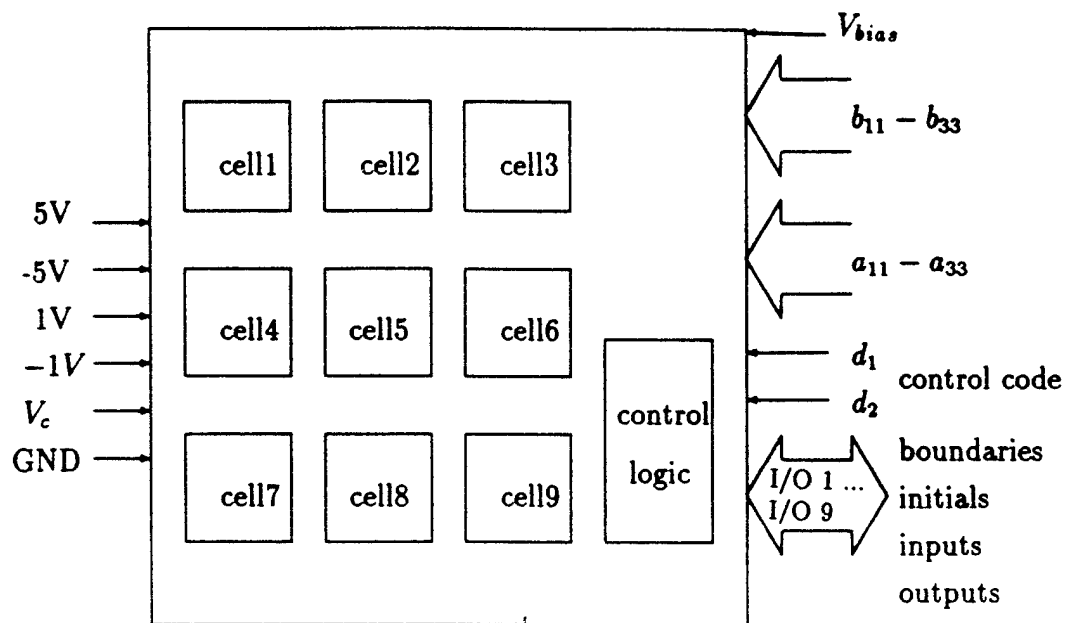
Figure 1. The system structure of the 3 × 3 CNN.
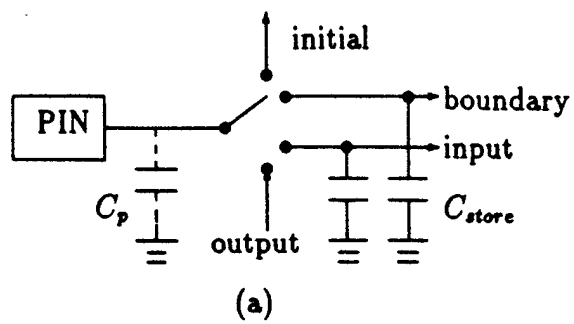
Figure 2. The diagram of time multiplexing switches to use the same pin for multi-purposes, C_p is the parasitic capacitance of the pin, and C_{store} is the storage capacitance.
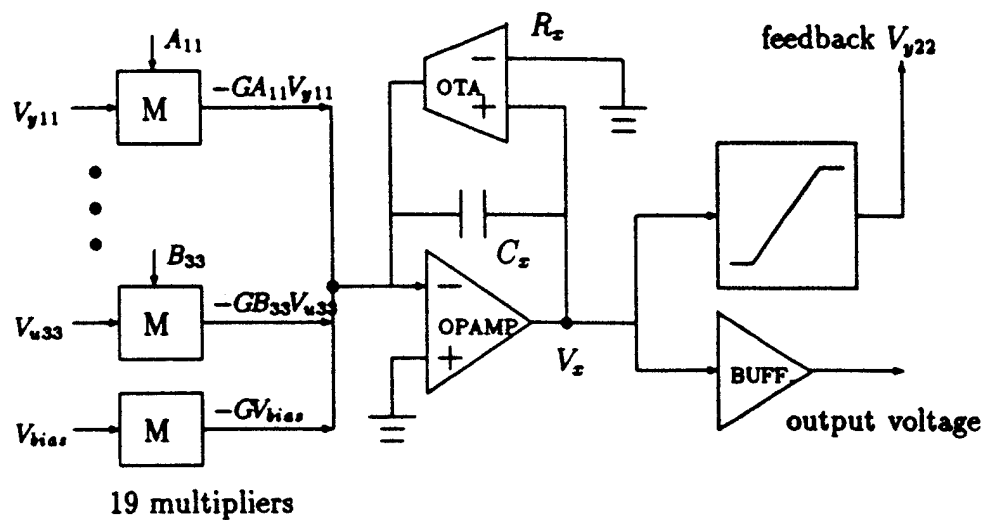
Figure 3. The building structure of the CNN cell. "M" means a transconductance multiplier with a current output.
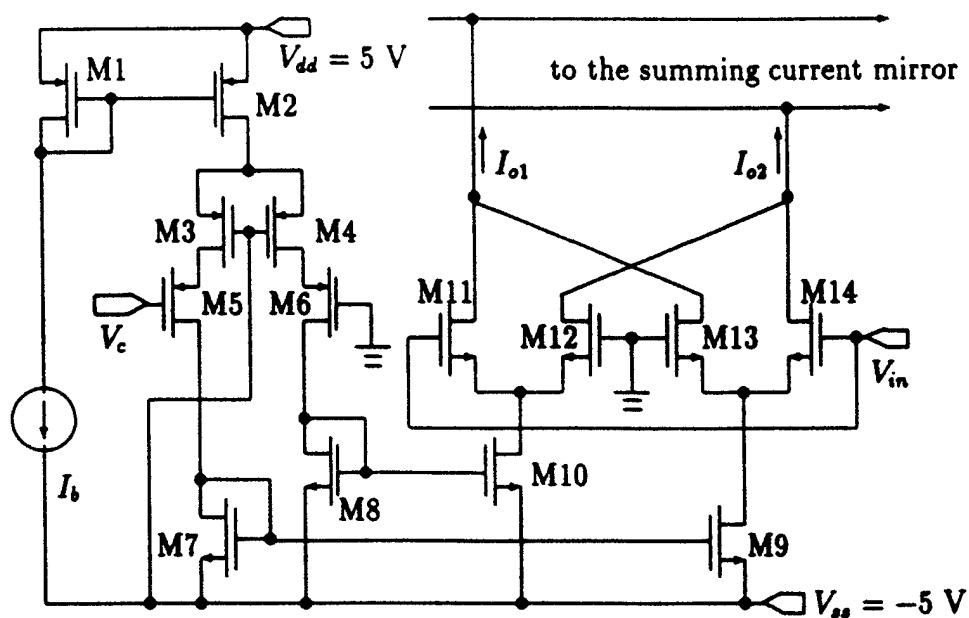
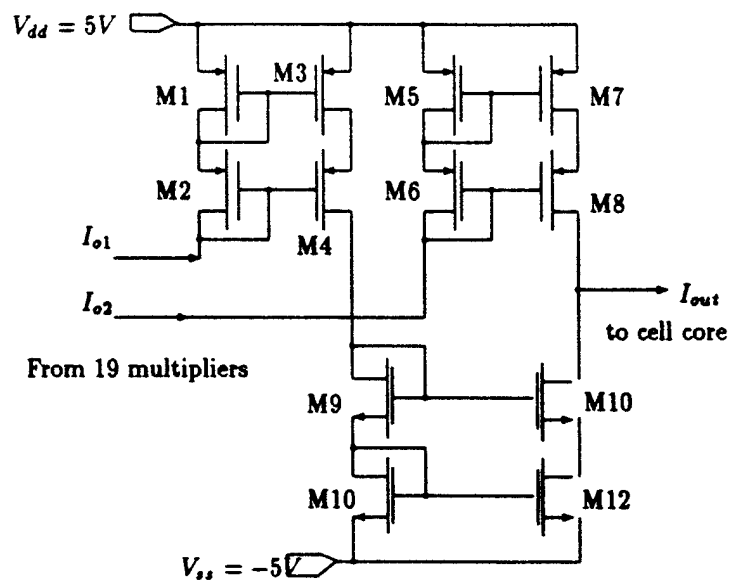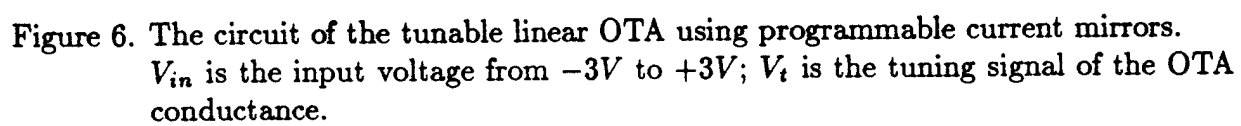Figure 4. Circuit of the folded Gilbert multiplier with linear expansion. V_c is the control voltage (temple input), and V_{in} is the image pixel input.

Figure 5. The summing current mirror of all 19 multipliers.

Figure 6. The circuit of the tunable linear OTA using programmable current mirrors. V_{in} is the input voltage from $-3V$ to $+3V$; V_t is the tuning signal of the OTA conductance.
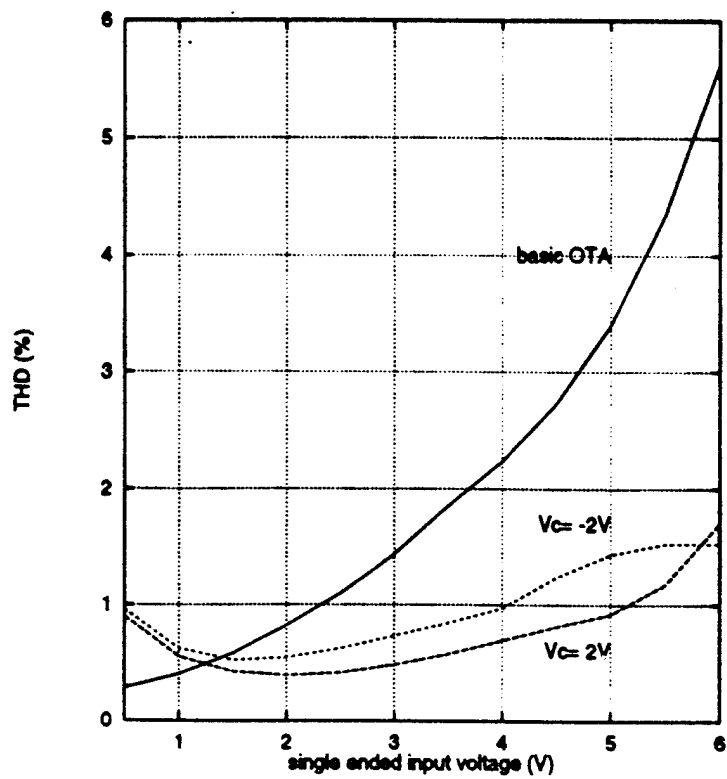
Figure 7. Plots of THD vs. input voltages (peak–to–peak value) of the "basic" OTA and the tunable OTA at $V_t = -2V$ and $V_t = 2V$.
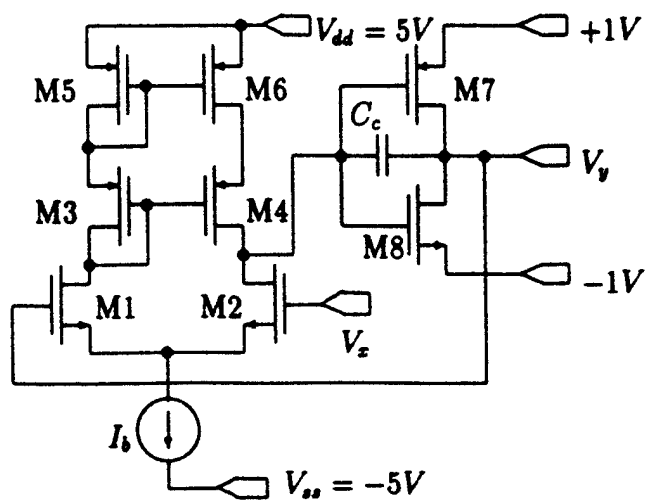
Figure 8. The circuit structure of the sigmoid activation function, an opamp having separate power voltages and in the unit feedback connection.
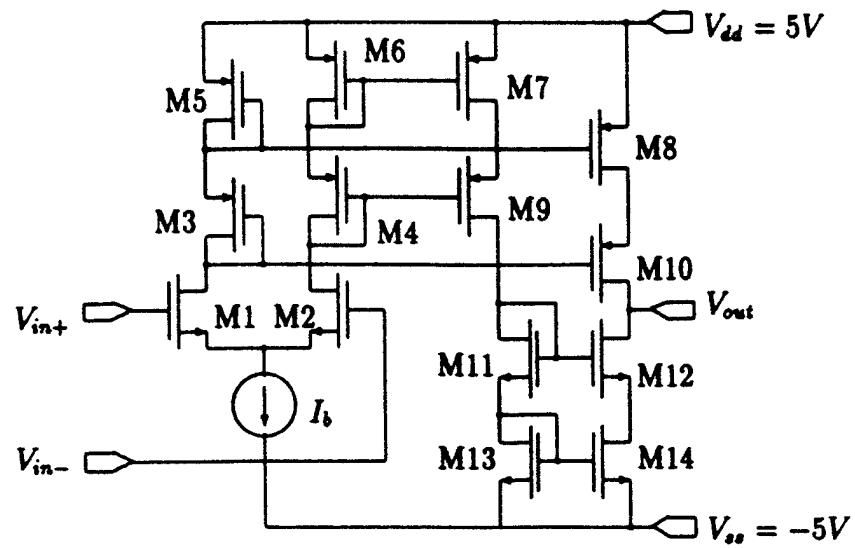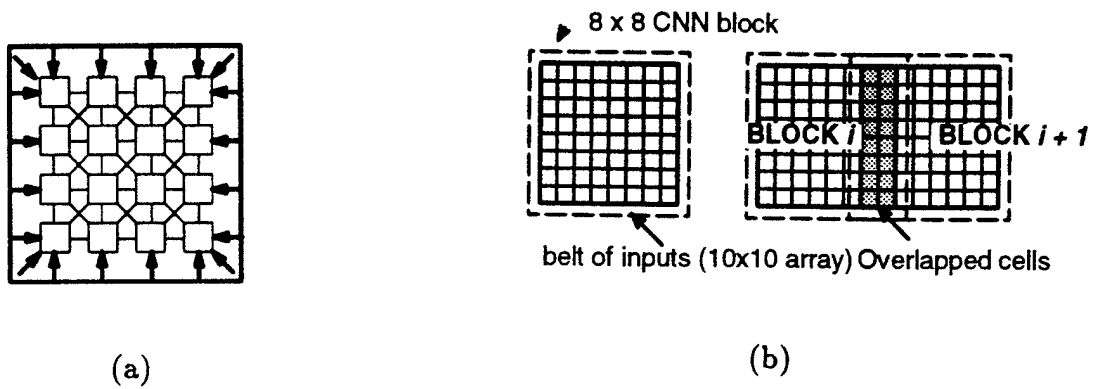
Figure 9. The circuit structure of the one–stage opamp.

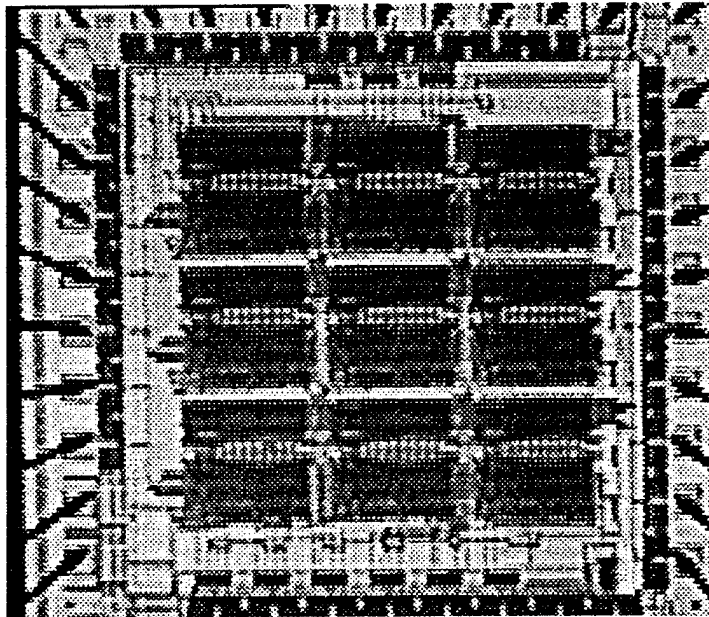Figure 10. Conditions for Time Multiplexing Opeation; (a) Belt of Inputs, (b) Overlapped Pixels

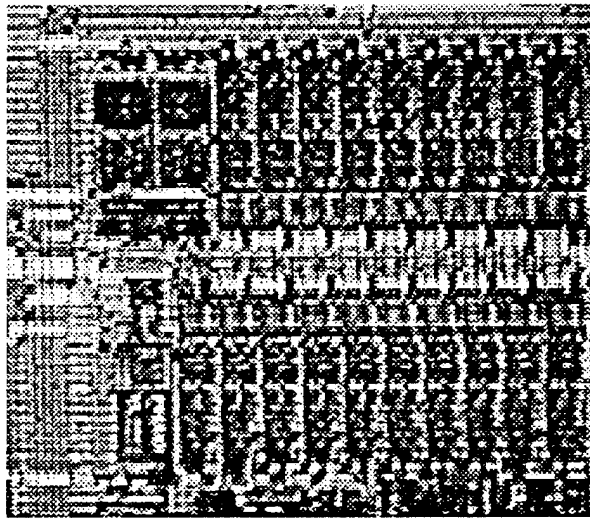Figure 11a. The photograph of the die of the 3 × 3 CNN.
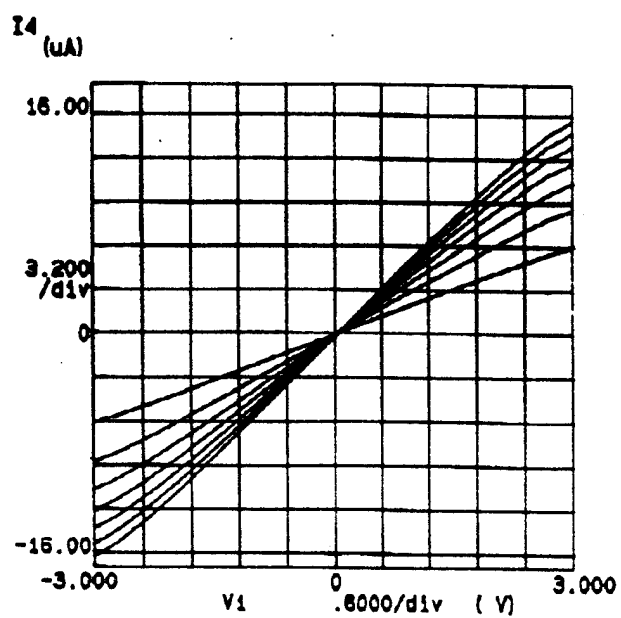
Figure 11b. The photograph of the circuit of one cell.

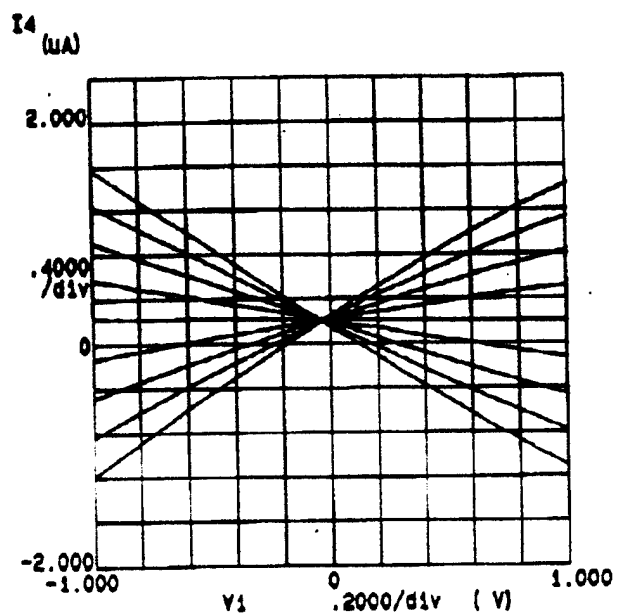Figure 12. The DC sweep characteristic of the tunable OTA, where the tuning voltage V_c is from $-2.0V$ to $+2.0V$.

Figure 13. The DC seep characteristics of the analog multiplier, where the sweep voltage
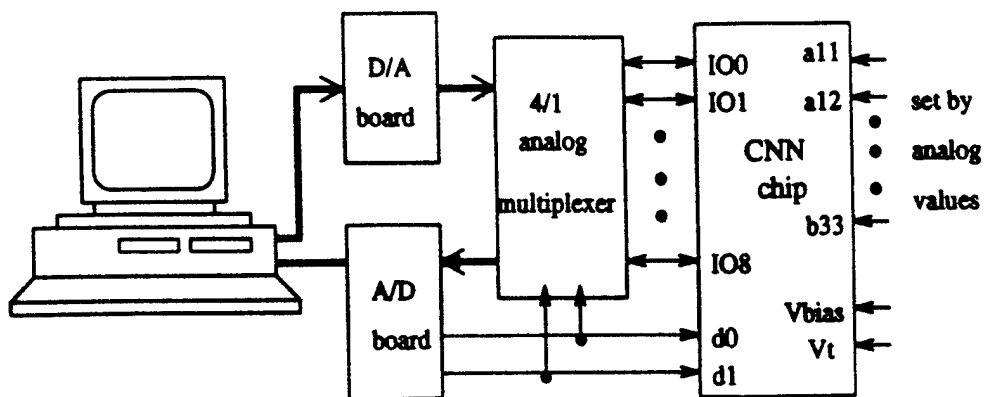if from $-1.0V$ to $+1.0V$.

Figure 14. The general interfacing system of the CNN chip with a person computer.
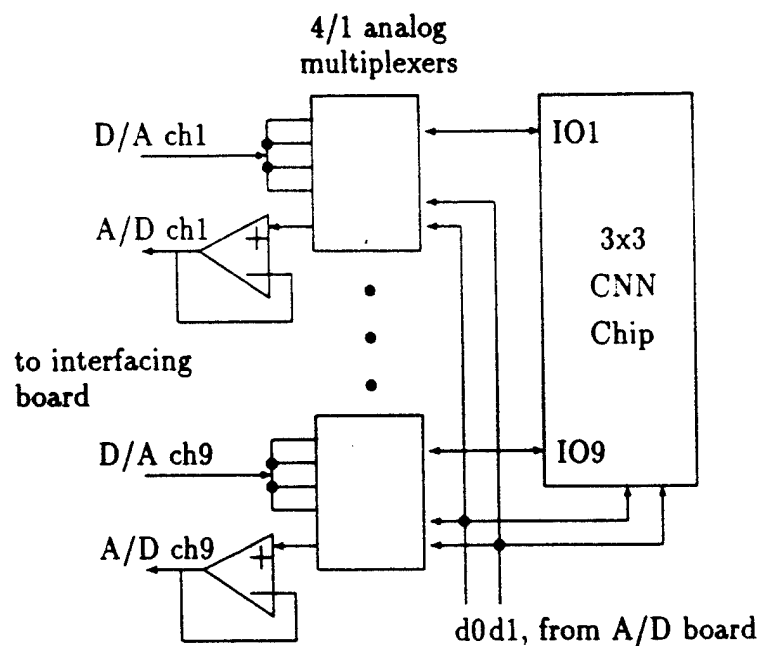

Figure 15. The detailed connections of the 4-1 analog multiplexers outside the CNN chip.

Figure 16a. The unprocessed 256 × 256 input images (gray level image).

Figure 16b. The unprocessed 256 x 256 input images (color image).

Figure 17a. The output image after applying the edge detection template (gray level image).

Figure 17b. The output image after applying the edge detection template (color image).

Figure 18a. The output images after applying the hole filling template(gray level image).

Figure 18b.   The output images after applying the hole filling template (color image).

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE November 15, 1995	3. REPORT TYPE AND DATES COVERED Technical Report 08/01/95-10/01/95

4. TITLE AND SUBTITLE

TIME MULTIPLEXED COLOR IMAGE PROCESSING BASED ON A VLSI CELLULAR NEURAL NETWORK WITH CELL-STATE OUTPUTS

5. FUNDING NUMBERS

N99914-94-1-0516

6. AUTHOR(S)

Jose Pineda de Gyvez

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Texas Engineering Experiment Station
Texas A&M University

8. PERFORMING ORGANIZATION REPORT NUMBER

93-549/#7

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
Cose 214:JWK
Ballston Tower One
800 North Quing Street
Arlington, Virginia 22217-5660

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

A: Approved for public release:distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report presents a Cellular Neural Network (CNN) VLSI implementation to be used in a time–multiplexing scheme for processing large and complex images. While efficient VLSI implementations have been reported, no research work has addressed the use of small CNN arrays for processing large complex images. Hence, the work hereby presented introduces a strategy to process large images using small CNN arrays. For practical size applications, due to hardware limitations, it is impossible to have a one–on–one mapping between the CNN hardware cells and all the pixels in the image involved. This report presents a practical solution by processing the input image block by block, with the number of pixels in a block being the same as the number of CNN cells in the array. Furthermore, unlike other implementations in which the output is taken from the feedback path of each cell, the VLSI architecture hereby described takes the outputs from the state node. While previous implementations are mostly suitable for black and white applications because of the thresholded outputs, our approach is especially suitable for applications in color (gray) image processing due to the analog nature of the state node.

14. SUBJECT TERMS

CELLULAR NEURAL NETWORKS, VLSI, MULTIPLEXING

15. NUMBER OF PAGES

41

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std. 239-18

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet *optical scanning requirements.*

Block 1. Agency Use Only *(Leave blank).*

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	-	Contract	PR	- Project
G	-	Grant	TA	- Task
PE	-	Program	WU	- Work Unit
		Element		Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

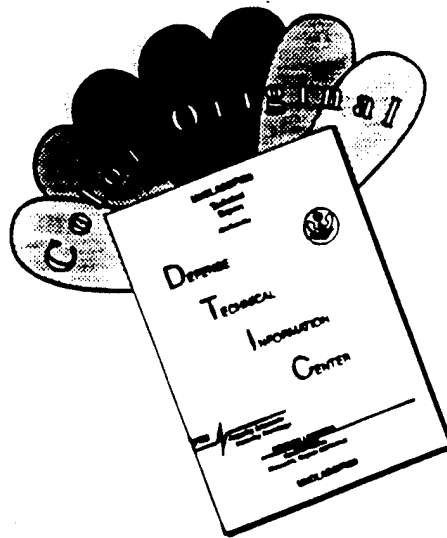**Block 14.** Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only).*

Blocks 17.- 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

DISCLAIMER NOTICE